VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXPERIMENTÁLNÍ PŘEKLADAČ Z ČEŠTINY DO SLOVENŠTINY

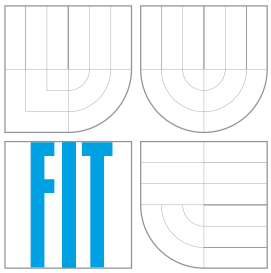
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ ZACHAR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXPERIMENTÁLNÍ PŘEKLADAČ Z ČEŠTINY DO SLOVENŠTINY

CZECH-SLOVAK MACHINE TRANSLATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ ZACHAR

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2012

Abstrakt

Tahle práce popisuje princip strojového překladu, oboznamuje čtenáře s již existujícimi systémem pro strojový překlad Moses a navrhuje systém, kteří za jeho pomoci je schopen se naučit a posléze překládat text z češtiny do slovenštiny.

Abstract

This thesis describes ideas and theories behind machine translatinon, informs the reader about existing machine translation system Moses and by utilizing it proposes system, which is able to learn and later translate from Czech language into Slovak language.

Klíčová slova

strojový překlad, zarovnání slov, korpus

Keywords

machine translation, word alignment, corpus

Citace

Lukáš Zachar: Experimentální překladač z češtiny do slovenštiny, diplomová práce, Brno, FIT VUT v Brně, 2012

Experimentální překladač z češtiny do slovenštiny

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana docenta Pavla Smrže. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal

.....
Lukáš Zachar
23. května 2012

Poděkování

Ďakujem môjmu vedúcemu diplomovej práce za trpezlivosť a odbornú pomoc. Taktiež by som sa rád poďakoval všetkým, ktorí mi akokoľvek pomohli pri tvorbe tejto práce, či už odborne alebo ľudsky.

Obsah

Kapitola 1

Úvod

Ľudstvo už od čias Babylonu bojuje s problémom vzájomného nedorozumenia. Tlmočníci, ktorí dokázali sprostredkovať komunikáciu medzi navzájom odlišne hovoriacimi osobami boli, sú a budú vždy vyhľadávaní.

Začiatky rozvoja strojového prekladu siahajú do 50. rokov 20. storočia [1], v časoch studenej vojny bolo nutné rozumieť druhej strane a keďže prvé pokusy so strojovým prekladom dopadli nad očakávania, všetci boli plní nadšenia, že funkčné systémy budú hotové čoskoro. Ale nestalo sa tak.

Dochádzalo k ojedinelým úspechom v značne obmedzených oblastiach ako preklad predpovedí počasie a podobne.

Z postupným nárastom výpočtovej sily počítačov, lacnejšími veľkokapacitnými pamäťami a pravdepodobne hlavne s vytvorením celosvetovej siete začal veľký nárast výskumu v tejto oblasti.

Od modelov postavených na jazykovedných princípoch sa vývoj začal uberať do štatistického a na príkladoch založeného prekladu, ktoré začínajú dosahovať úspechy.

Štruktúra tejto práce je nasledovná:

Kapitola 2 je úvodom do teoretického pozadia strojového prekladu. Najprv je v nej vysvetlený spôsob zarovnania viet, ktorý je dôležitý z hľadiska ďalšieho spracovania. Potom nasleduje vysvetlenie akým spôsobom preklad pracuje, od jednoduchých modelov IBM Model 1 až po modely založené na stromových štruktúrach. Hovorí sa v nej aj o spôsobe symetrizácie zarovnania slov.

Nasledujúca kapitola 3 nastoľuje opis systému, ktorý by mal byť schopný preložiť text zo češtiny do slovenčiny. Pretože strojový preklad priamo závisí na dátach nad ktorými bol naučený, začína táto kapitola uvedením možných zdrojov dát. Nasleduje opis použitého systému a jeho jednotlivé kroky.

Kapitola 4 je venovaná poznatkom a experimentom získaných počas tvorby systému.

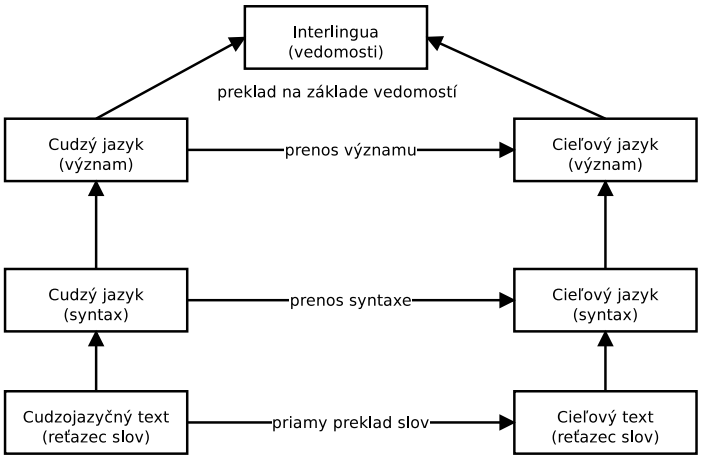
Posledná kapitola 5 rekapituluje, čo bolo vykonané v tejto práci, čo malo byť vykonané inak a čím by mohla táto práca pokračovať do budúcnosti.

Text tejto práce je nevychádza zo semestrálneho projektu, žiadna jeho časť nebola znovu použitá.

Kapitola 2

Teoretické pozadie strojového prekladu

Existuje viacero rôznych prístupov k automatickému prekladu, ktoré pracujú na rôznych úrovniach. Schéma tých najdôležitejších prístupov je na obrázku 2.1.



Obrázok 2.1: Vauqoisov trojuholník [9]

Najnižšou úrovňou je skoro mechanický preklad slov na základe slovníka. Tento prístup neuvažuje žiadne závislosti medzi slovami a preto je jeho výstup často neprirodzený a veľakrát sa stráca pôvodný zmysel vety.

Druhá úroveň sa snaží zachytiť vzťahy medzi slovami pomocou odhalenia syntaktických štruktúr viet. Problémom je, že táto úloha častokrát nie je jednoznačná, najmä pokiaľ sa nesnažíme pochopiť zmysel vety. Výstupom už je text, ktorý je možné považovať za gramaticky správny, ale ešte stále dochádza k zmene významu vety.

Treťou úrovňou hľadáme zmysel vety, ktorý doslovne preložíme do výstupného jazyka. Hoci zmysel vety zostáva zachovaný, poradie slov môže byť neprirodzené, lebo rešpektuje

vetnú skladbu pôvodného jazyka.

Poslednou úrovňou je porozumenie vete a následný vysvetlenie do cieľového jazyka. Tento prístup by v prípade viacnásobného prekladu, alebo výmeny dokumentov medzi viacerými stranami, značne zjednodušil celkovú náročnosť tvorby modelov. Takto by bolo dostačujúce pre každý zúčastnený jazyk vytvoriť model pre preklad z a do spoločnej reprezentácie znalostí. I keď sa do tvorby podobných systémov vynaložilo značné úsilie, realizácia tohto prístupu je ešte stále veľmi vzdialená.

Podstatne schodnejším prístupom sa ukázalo využitie metód strojového učenia pre tvorbu prekladových modelov. Obzvlášť s postupným prenikaním veľkého množstva textov na internet, ktoré je možné spracúvať a nechať stroje vytvárať svoje vlastné modely.

2.1 Zarovnanie viet

Zdrojom dát pre tvorbu prekladových modelov sú viacjazyčné texty. Pochádzajú z prostredia, v ktorom je nutné zachovať myšlienku a distribuovať ju príjemcom hovoriacich rozdielnymi jazykmi. Viac informácií o najviac využívaných zdrojoch týchto textov bude neskôr uvedených v 3.1.

Pretože prekladové modely uvažujú vetu ako základ ďalšieho spracovania textu, potrebujeme získať informáciu o tom, ktoré vety vyjadrujú tie isté myšlienky. V ideálnom prípade každej vete jedného jazyka odpovedá veta druhého jazyka, ale častokrát je táto väzba porušená a prekladateľ sa rozhodol rozdeliť alebo spojiť text.

Zarovnanie viet je problémom, v ktorom hľadáme spojenie medzi skupinami viet oboch textov na základe obsahu, ktorý vyjadrujú. Jeho riešenie je o to ťažšie, že nie sme schopní porozumieť tomu čo vyjadrujú.

Prvé algoritmy pre zarovnávanie textu vychádzali z porovnávania dĺžky viet. Myšlienkou tohto postupu je, že kratšia veta by mala zostať kratšou aj po svojom preložení. Podobne to očakávame aj pre vety dlhšie.

Gale a Church

Navrhli jednu z prvých štatistických metód pre určenie zarovnania medzi vetami. Metóda sa snaží nájsť zarovnanie A, ktoré dosahuje najväčšiu pravdepodobnosť pre danú dvojicu textov S a T [4].

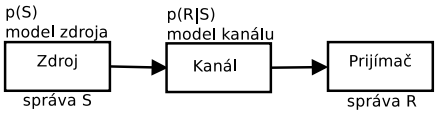
Text je rozložebý do viacerých súvislých blokov $(B_1, \ldots, B_k)$, pričom žiadna veta nemôže patriť do dvoch blokov. Potom môžeme výslednú pravdepodobnosť zarovnania vyjadriť ako:

$$P(A|S,T) \approx \prod_{k=1}^{K} P(B_k) \tag{2.1}$$

Gale a Church uvažovali iba niektoré, najčastejšie, možnosti zarovnania medzi vetami (1:1,1:0,0:1,2:1,1:2,2:2) a tým umožnili určiť najpravdepodobnejšie zarovnanie využitím dynamického programovania. Aby táto metóda mohla fungovať je nutné určiť cenu zarovnania uvažovaných skupín viet. Výpočet ceny je založený na počte znakov, ktoré dané skupiny viet obsahujú. Predpokladom je, že každý znak skupiny viet jedného jazyka zapríčiní výskyt náhodného počtu znakov viet druhého jazyka. Tieto náhodné premenné sú modelované ako Gaussovo rozloženie, ktorého parametre sú určené na základ existujúceho korpusu pre daný pár jazykov. Samotná cena zarovnania je určená pomocou metriky vzdialenosti medzi súčtami počtov znakov medzi dvomi blokmi. Táto metóda dosahuje spoľahlivé výsledky pre párovanie 1:1, ale pri vynechaných párach (1:0, 0:1) nikdy nedosiahne správny výsledok.

2.2 Princíp prekladu

Preklad textu je založený na princípe, že z vety v cudzom jazyku **f** by sme radi vytvorili vetu v cieľovom jazyku **e**. Jedným zo spôsobov, ako túto činnosť zaradiť do kontextu, v ktorom môžeme využiť existujúce poznatky z teórie informácií, je vyjadriť daný problém pomocou modelu Kanálu so šumom (Noisy-channel, obrázok 2.2).



Obrázok 2.2: Kanál so šumom [6]

Myšlienkou je, že veta bola pôvodne v cieľovom jazyku, ale počas prenosu došlo k jej zmene do cudzieho jazyka a až takto modifikovanú sme sa ju dozvedeli. Naším cieľom je tak nájsť pôvodný text prijatej vety.

Typický systém pozostáva z troch častí:

Model jazyka ktorý vyjadruje pravdepodobnosť, s akou je daná veta skutočnou vetou v danom jazyku. Túto pravdepodobnosť vyjadrujeme ako $P(e)$.

Model prekladu ktorý vyjadruje pravdepodobnosť, s akou je daná veta cieľového jazyka prekladom prekladom vety v cudzom jazyku. Túto pravdepodobnosť vyjadrujeme ako $p(f|e)$

Dekodér ktorý spája oba modely a hľadá takú vetu cieľového jazyka, pre ktorú dosahuje najväčšiu pravdepodobnosť. Jeho činnosť možno vyjadriť ako:

$$\begin{aligned} \arg\max_e p(\mathbf{e}|\mathbf{f}) &= \arg\max_e \frac{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})}{p(\mathbf{f})} \\ &\sim \arg\max_e p(\mathbf{f}|\mathbf{e})p(\mathbf{e}) \end{aligned} \tag{2.2}$$

Táto časť je výpočtovo najzložitejšia a je jadrom celého systému.

V nasledujúcich sekciách budú postupne bližšie predstavené jednotlivé princípy, pomocou ktorých môžme implementovať jednotlivé časti.

2.2.1 Modely založené na slovách

Patria k najstarším spôsobom prekladu a v súčastnosti sa už v tejto forme nepoužívajú. Avšak nimi objavené koncepty prekladu umožnili vznik sofistikovanejších modelov [6].

Najjednoduchší je model, ktorý je založený na prekladovom slovníku. Napr. **kolo** - *koleso, bicykel.* Ako už aj z príkladu možno zbadať, väčšina slov nemá jednoznačný ekvivalent v druhom jazyku. Využitím štatistických metód je možné jednotlivým dvojiciam výrazov priradiť ich pravdepodobnosť. Vytvoríme funkciu

$$p_f : e \rightarrow p_f(e) \tag{2.3}$$

ktorá cudziemu slovu f priradí pravdepodobnosť jeho prekladu e.

Pretože poradie slov môže byť rozdielne a tiež aby tento model bol schopný spracovať viacslovné výrazy, ktoré môžu mať v obidvoch jazykoch rozdielny počet slov, je potrebné definovať zarovnanie medzi jednotlivými slovami. Toto zarovnanie definuje funkciu

$$a : i \rightarrow j \tag{2.4}$$

ktorá každému indexu slova v jednom jazyku priradí index slova v druhom jazyku. V niektorých prípadoch slovo nemá svoj ekvivalent v druhom jazyku (napríklad zdôraznenie anglického „do“), preto je zavedený token NULL, ktorý chápeme ako prázdne slovo.

IBM Model 1

Je prvým a najjednoduchším príkladom modelu založeného na pravdepodobnosti doslovných prekladov. Veta cieľového jazyka je tvorená slovami $e = (e_1, ..., e_{l_e})$, pôvodná veta v cudzom jazyku slovami $f = (f_1, ..., f_{l_f})$. Zarovnanie jednotlivých slov je určené pomocou funkcie a. Výsledná pravdepodobnosť prekladu je vyjadriteľná pomocou vzťahu

$$p(\mathbf{e}, a|\mathbf{f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) \tag{2.5}$$

Podobne ako v ostatných oblastiach aplikácie strojového učenia využívame pre získanie modelu prekladu, teda pravdepodobnosti prekladu dvojíc slov, algoritmus Expectation-Maximalization. Tento algoritmus pracuje na nasledujúcom princípe:

1. Priraď náhodné (rovnomerné) rozloženie pravdepodobnosti.
2. Aplikuj existujúci model na dáta.

3. Uč sa z modelu nové rozloženie pravdepodobnosti.

4. Opakuj kroky 2 a 3 až kým nenastane konvergencia.

Aplikáciou modelu na dáta sa v prípade IBM Modelu 1 rozumie výpočet pravdepodobnosti zarovnania a pokiaľ máme daný pár viet $\mathbf{e}, \mathbf{f}$.

$$p(a|\mathbf{e},\mathbf{f}) = \frac{p(\mathbf{e},a|\mathbf{f})}{p(|\mathbf{e}|\mathbf{f})} = \prod_{j=1}^{l_e} \frac{t(e_j|f_{a(j)})}{\sum_{i=0}^{l_f} t(e_j|f_i)} \tag{2.6}$$

Následne je potrebné vykonať ďalší krok algoritmu EM a to úprava modelu. Zavedieme funkciu c pre určenie počtu prípadov, v ktorých je konkrétne slovo f preloží pomocou slova e:

$$c(e|f;\mathbf{ef}) = \sum_a p(a|\mathbf{ef}) \sum_{j=1}^{l_e} \delta(e,e_j)\delta(f,f_{a(j)}) \tag{2.7}$$

Funkcia $\delta(i,j)$ je Kroneckerovo delta, ktoré má hodnotu 1 pre $i = j$ a 0 v ostatných prípadoch.

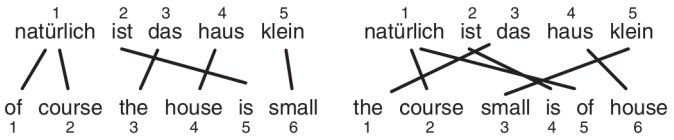
Využitím funkcie počtu prekladov sa nová pravdepodobnosť prekladu týchto dvoch slov vypočíta ako

$$t(e|f;\mathbf{ef}) = \frac{\sum_{(\mathbf{ef})} c(e|f;\mathbf{ef})}{\sum_e \sum_{(\mathbf{ef})} c(e|f;\mathbf{ef})} \tag{2.8}$$

Tento IBM Model 1 je príliš jednoduchý a je nutné ho rozšíriť o ďalšie vlastnosti. Jeho hlavným problémom je, že prázdny reťazec považuje za najlepší preklad. Taktiež považuje všetky zarovnania slov za rovnako pravdepodobné, čo nerešpektuje reálnu situáciu.

IBM Model 2

Tento model adresuje problém so zarovnávaním. V predchádzajúcom modely sú obe zarovnania znázornené na obrázku 2.3 považované za zhodné. Je to spôsobené neexistenciou explicitného modelu pre zarovnanie slov.



Obrázok 2.3: Chybné vyhodnotenie zarovnania IBM Modelu 1[6]

IBM Model 2 tento problém odstraňuje pridaním pravdepodobnostného modelu pre zarovnanie slov 2.9. Pravdepodobnosť zarovnania závisí od pozície oboch slov vo vete.

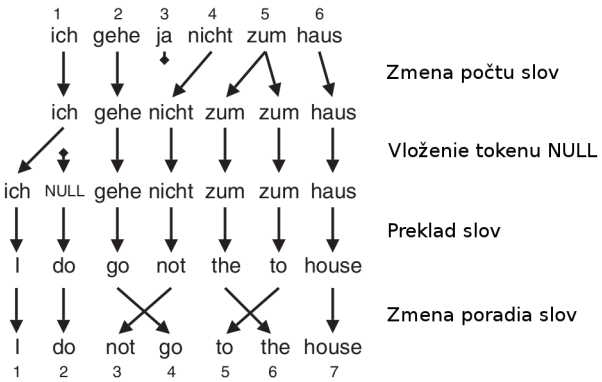
$$a(i|j,l_e,l_f) \tag{2.9}$$

IBM Model 3

Tento model pridáva možnosť modelovania skutočnosti, že počet slov pôvodného a preloženého textu môže byť rôzny. Predchádzajúce modely predpokladali zhodný počet slov. Množstvo generovaných slov cieľového jazyka z konkrétneho cudzieho slova je modelované pravdepodobnostnou distribúciou 2.10.

$$n(\phi|f) \tag{2.10}$$

Pretože prípustným počtom generovaných slov je aj 0, je umožnené tým tiež vyjadriť pravdepodobnosť, že pre konkrétne slovo neexistuje preklad do cieľového jazyka. Pripustenie možnosti zmeny počtu slov ale komplikuje už zavedený koncept tokenu NULL, ktorý dovoľuje vyjadriť pravdepodobnosť vloženie slova. Dôvodom je fakt, že pravdepodobnostná funkcia by považovala všetky pozície NULL za rovnocenné. Preto krok s generovaním tokenov NULL nastupuje až po vyhodnotení pravdepodobnosti pre zmazanie alebo znásobenie slov 2.4.



Obrázok 2.4: Kroky prekladu IBM Modelu 3 [6]

Hoci sa posledný krok premenoval zo zarovnania slov na zmenu poradia slov, je modelovaný podobným spôsobom ako v IBM Modele 2. Zmenený je len smer modelovania presunu a to v smere prekladu. Obe tieto zmeny zohľadňujú skutočnú úlohu tohoto kroku, ktorou je uviesť preložené slová do poradia rešpektujúceho daný jazyk.

Samotný výpočet pravdepodobnosti prekladu vety je vyjadrený ako:

$$
\begin{aligned}
p(\mathbf{e}|\mathbf{f}) &= \sum_{a} p(\mathbf{e}, a|\mathbf{f}) \\
&= \sum_{a(1)=0}^{l_f} \cdots \sum_{a(l_e)=0}^{l_f} \left(\begin{array}{c} l_e - \phi_0 \\ \phi_0 \end{array} \right) p_1^{\phi_0} p_0^{l_e - 2\phi_0} \prod_{i=1}^{l_f} \phi_i! n(\phi_i|f_i) \\
&\quad \times \prod_{j=1}^{l_e} t(e_j|f_{a(j)}) d(j|a(j), l_e, l_f)
\end{aligned} \tag{2.11}
$$

V tomto modele je už vyhodnocovanie všetkých možností zarovnania nereálne a je nutné sa uchýliť sa k použitiu heuristických metód. Za počiatočný stav môžeme použiť zarovnanie, ktoré získame vyhodnotením pomocou IBM Modelu 2 a následným využitím Gradientného algoritmu nájdeme lokálne maximá. Pre použitie tejto metódy je potrebné definovať susediace zarovnania: Sú to také zarovnania, ktoré sa líšia len posunom alebo zámenou.

Posun Zarovnania a_1 a a_2 sa líšia posunom, pokiaľ je rozdiel v zarovnaní len jedného slova.

Zmena Zarovnania a_1 a a_2 sa líšia zámenou, pokiaľ je jediným rozdielom medzi ich zarovnaniami zámena zarovnania dvoch slov.

IBM Model 4

Hoci už predchádzajúci model zahŕňa väčšinu krokov potrebných pre správnu funkciu prekladu, trpí problémom so zámenou slov. Obzvlášť pri dlhých vetách dochádza k produkcii príliš vzdialených a nerealistických presunov slov.

Preto IBM Model 4 zaviedol relatívny presun slov, ktorý závisí od predošlých presunov slov. Slov, ktoré patria to tej istej skupiny zarovnania, tvoria tzv. cept. Pre každú takúto skupinu zavedieme pojem centra ceptu $\odot$, ktoré vyjadruje zaokrúhlený priemer súčtov pozícií, na ktorých sa nachádzajú slová patriace do preloženého textu. Relatívny posun je určený na základe hodnoty stredu ceptu $\odot$ a počiatočného indexu ceptu v pôvodnom texte π.

Pretože stále zachovávame modelovanie presunu pomocou pravdepodobnosti, je vhodné rozlíšiť medzi jednotlivými skupinami slov. Dôvodom je skutočnosť, že napríklad predložky majú zvyčajne iný posun ako prídavné mená. Riešením je zaviesť funkcie $\alpha(f)$ a $\beta(e)$, ktoré určujú do ktorej triedy cudzie a preložené slovo patrí. Pokiaľ nemáme k dispozícii nástroje, ktoré by nám určili triedy slov, môžeme automaticky rozdeliť všetky slová do dopredu pevne daného počtu tried.

$$
\begin{aligned}
\text{prvé slovo cept-u:} &\quad d_1(j - \odot_{i-1}|\alpha(f_{i-1}), \beta(e_j))) \\
\text{ďalšie slová:} &\quad d_{>1}(j - \pi_{i,k-1}|\beta(e_j)))
\end{aligned} \tag{2.12}
$$

Postup učenia IBM Modelu 4 je veľmi podobný predošlému modelu.

IBM Model 5

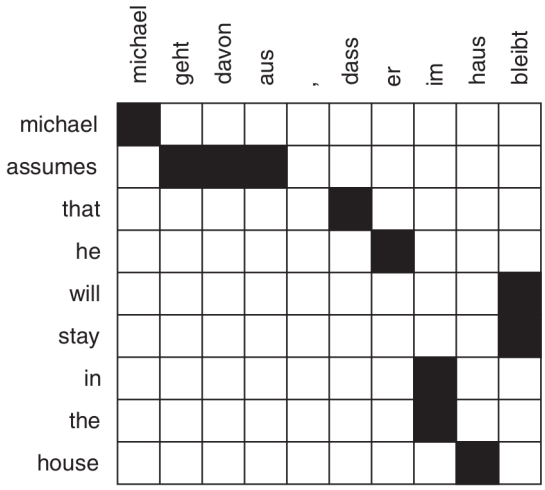
Je posledným z modelov a hlavným dôvodom pre jeho zavedenie je odstránenie konfliktu medzi viacerými možnými zarovnaniami. V predchádzajúcich modeloch bolo možné, aby viacero slov bolo presunutých na tú istú pozíciu. Pretože využívame algoritmus EM pre znovu naučenie rozdelenia, je možnosť takýchto konfliktov v pre naše účely kontraproduktívna.

Preto IBM Model 5 udržuje zoznam pozícií, ktoré ešte neboli obsadené a v poslednom kroku dovoľuje obsadiť iba doposiaľ voľné pozície. Zmení sa tak model obsadzovania pozícií 2.13. V každom kroku je nutné získať informáciu o tom, ktoré pozície sú ešte voľné v

$$\begin{aligned} \text{prvé slovo cept-u:} \quad & d_1(v_j|\beta(e_j), v_{\odot_{i-1}}, v_{max}) \\ \text{ďalšie slová:} \quad & d_{>1}(v_j - v_{\pi_{i,k-1}}|\beta(e_j), v_{max}) \end{aligned} \tag{2.13}$$

2.2.2 Zarovnanie slov medzi vetami

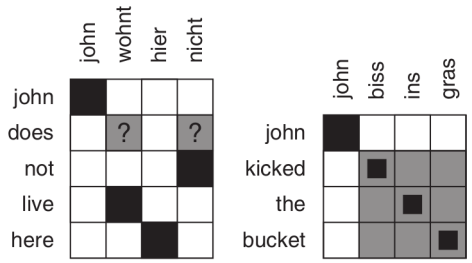
Dôležitým konceptom, ktorý zaviedli v prechádzajúcej časti definované IBM modely, je zarovnanie slov medzi dvomi vetami. Toto zarovnanie je zvyčajne reprezentované ako matica zarovnania, v ktorej stĺpce a riadky predstavujú slová jednotlivých viet a podľa položky indexovanej týmito slovami je možné určujú, ktoré kombinácie slov sú si navzájom zarovnané 2.5.



Obrázok 2.5: Matica zarovnania slov [6]

Zarovnanie nie je vždy jednoznačné, okrem pomocných slov je problematické aj výsledné zarovnanie ustálených fráz, ktoré je potrebné ponechať v celku. Problém je ilustrovaný na obrázku 2.6. Príklad s pomocným slovesom *do* v ľavej časti obrázku vyjadruje nejednoznačnosť priradenia. Sú tri možnosti zarovnania a každé je zdôvodniteľné:

- Slovo nemá ekvivalent v nemčine a preto ho nepriradíme k žiadnemu slovu.
- Slovo vyjadruje informáciu o čase a osobe hlavného slovesa *live*, preto je priradené k ekvivalentnému slovesu *wohnt* v nemčine.
- Slovo bolo zavedené z dôvodu negácie a preto by malo byť zarovnané k *nicht*.



Obrázok 2.6: Problém s určením zarovnania: vľavo naviazanie pomocných slovies, vpravo idiomatické spojenia. [6]

Príklad v pravej časti obrázku zas ilustruje nutnosť zachovať celú idiomatickú frázu ako celok. Žiadne iné zarovnanie by nedávalo správny preklad (napr. "*bucket* (vedro) a *gras* (tráva)).

Pretože kvalita zarovnania slov je jedným z predpokladov pre správnosť vytvoreného modelu, existuje záujem o vytvorenie čoraz lepších metód pre zarovnanie slov. Aby bolo možné určiť, ktorá metóda je lepšou, bola manuálne vytvorená referenčná množina zarovnaní, na základe ktorej je možné určiť kvalitu zarovnania. Nevyrieši sa tým ale nejednoznačnosť niektorých prípadov. Preto sú body prvky zarovnania rozdelené do dvoch kategórií: isté a možné. Často používanou metrikou na vyhodnotenie zarovnania slov je Chybovosť zarovnania (alignment error rate), ktorá je definovaná ako:

$$AER(S, P; A) = \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{2.14}$$

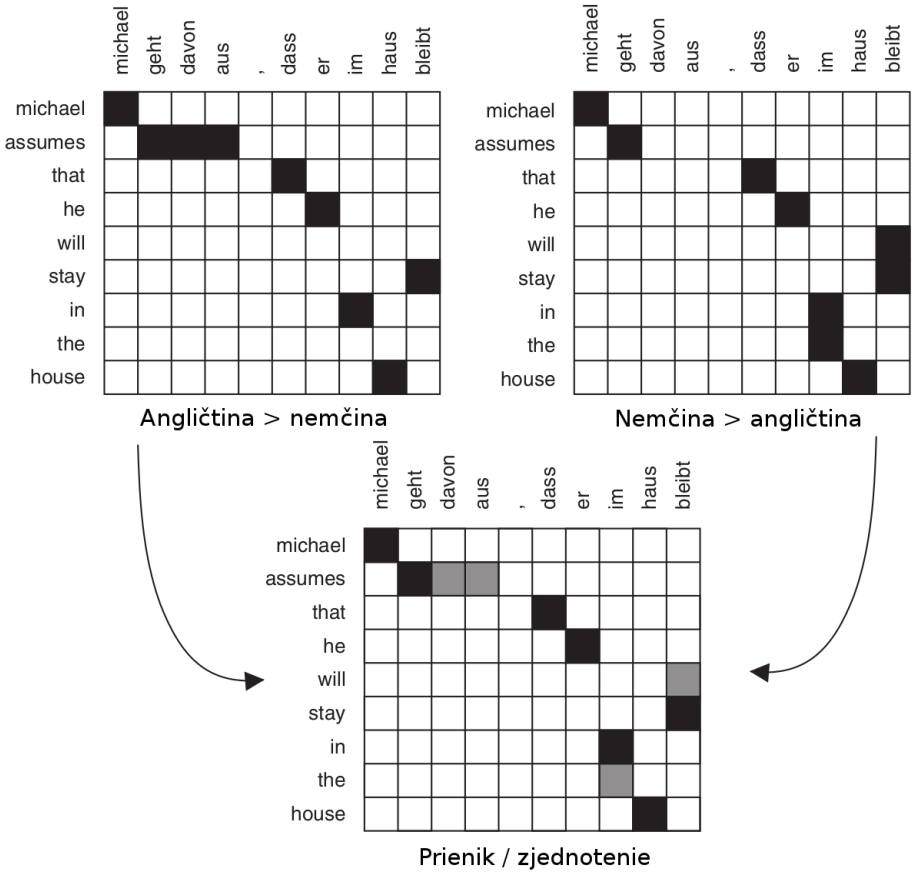
Metrika ohodnotí chybovosťou 0% také zarovnanie A, v ktorom boli nájdené všetky isté body zarovnania S a aspoň niektoré možné body zarovnania P.

Symetrizácia zarovnania slov

V praxi sa často používajú IBM modely pre určenie zarovnanie slov medzi vetami. Jedno z týchto zarovnaní je najpravdepodobnejšie a nazýva sa Viterbiho zarovnanie. Ako už bolo spomenuté, zarovnanie týchto modelov neumožňuje pokryť jedno slovo cieľového jazyka

viacerými slovami pôvodného textu, výsledné zarovnanie produkované týmito modelmi je preto vždy vo vzťahu 1 ku N. Preto použijeme IBM Model na vytvorenie tabuľky zarovnaní v oboch smeroch a následne ich spojíme (obrázok 2.7).

Existuje viacero možností ako spojiť oba nájdene zarovnania. Krajnými riešeniami sú použitie prieniku a zjednotenia nájdených prvkov. Prienik zaručene dosahuje vysokú mieru presnosti zarovnania lebo obsahuje len isté prvky. Problémom je, že ich neobsahuje všetky. Na druhej strane zjednotenie ic obsahuje všetky, ale okrem nich, čo je veľmi pravdepodobné, sa do výsledného zarovnania dostanú aj chybne nájdené prvky.



Obrázok 2.7: Symetrizácia zarovnania získaného z IBM Modelu [6]

Grow-diag-final Reálne používané prístupy preto ležia medzi týmito dvomi extrémami. Často používaným prístupom je metóda Grow-diag-final, založená na princípe pridávania susediacich prvkov zarovnania. Za susediace prvky sú považované všetky prvky 8-okolia. Algoritmus možno popísať pseudo kódom 2.8

```
GROW-DIAG-FINAL(e2f,f2e):
    susedia = ((-1,0),(0,-1),(1,0),(0,1),
                   (-1,-1),(-1,1),(1,-1),(1,1))
    zarovnanie = prienik(e2f,f2e);
GROW-DIAG(); FINAL(e2f); FINAL(f2e);

GROW-DIAG():
    rob kým je možné pridať nové prvky
        pre cieľové slovo e = 0 ... en
            pre cudzie slovo f = 0 ... fn
                ak ( e zarovnané s˜f )
                    for pre každý susediaci bod ( e', f' ):
                        if (( e' nie je zarovnaný alebo f' nie je zarovnaný )
                                a zároveň
                               ( e', f' ) patrí zjednoteniu ( e2f, f2e ) )
                            pridaj bod zarovnania ( e', f' )
FINAL(a):
    pre cieľové slovo e = 0 ... en
        pre cudzie slovo f = 0 ... fn
            if (( e' nie je zarovnaný alebo f' nie je zarovnaný ) a zároveň
                    ( e', f' ) patrí zjednoteniu ( e2f, f2e ) )
                pridaj bod zarovnania ( e', f' )
```

Obrázok 2.8: Pseudokód symetrizácie Grow-diag-final podľa [6]

Zarovnanie slov bez symetrizácie

Ako už bolo spomenuté, vo väčšine dnešných systémov sú používané IBM Modely pre nájdenie zarovnania slov. Pretože prvé dva modely (IBM Model 1, IBM Model 2) nedokážu modelovať znásobenie slov, sú podstatne rýchlejšie ako ich nástupníci. Pre väčšina súčasných systémov je ale presnosť tohto zarovnania nepostačujúca a musia využívať IBM modely, ktoré už dokážu modelovať aj znásobenie slov i za cenu podstatného spomalenia učenia. Naviac sú vrátené modely stále asymetrické a je nutné ich následne symetrizovať.

Zarovnanie dohodou Zaujímavý prístup navrhli vo svojom článku Liang a ďalší [8]. Snažili sa odstrániť nevýhody spojené s využívaním vyšších IBM modelov a navrhli spôsob, ako pomocou jednoduchšieho modelu založeného na princípe IBM Model 2 možno dosiahnuť lepšie výsledky zarovnania slov ako vzniknú pomocou symetrizácie výstupu vyšších modelov. Hlavnou myšlienkou je skutočnosť, že zhodné predpovede dvoch modelov predčia kvalitu každého z modelov zvlášť. Spôsob tvorby týchto modelov podporuje hľadanie dohody už počas trénovania.

Ich model zavádza závislosť pravdepodobnosti tokenu NULL od dĺžky cieľovej vety ako $p_0 = \frac{1}{l_e+1}$. Ďalej parametrizuje pravdepodobnosť presunu slova vyjadrení pomocou multinomického rozdelenia nad 11 hodnotami presunu $\leq -5, -4, \ldots, 4, \geq 5$ a uvažuje nad tromi

druhmi presunu: presun na začiatok, koniec a ostatné pohyby.

Aby bolo umožnené modelom nachádzať dohodu už počas trénovania tak využívajú tú istú tabuľku zarovnaní. Preto je aj pôvodne asymterické zarovnanie a nahradené indexom do tabuľky zarovnaní pre konkrétnu dvojicu slov. Oba modely tak vytvárajú vlastnú pravdepodobnostnú distribúciu nad rovnakými pármi viet a zarovnaní. Po natrénovaní oboch modelov pomocou EM algoritmu je potrebné určiť spôsob kombinácie týchto dvoch modelov pre predikciu zarovnania doposiaľ nevidených viet. Na jednej strane môžeme získať Viterbiho zarovnanie, alebo môžeme ponechať posteriálne dekódovanie, pri ktorom ponecháme hrany na základe vhodne zvolenej hraničnej hodnoty. Touto hodnotou môžeme priamo ovplyvňovať výslednú preferenciu presnosti alebo použitia (recall) modelu.

2.2.3 Modely založené na frázach

V súčasnosti patrí tento typ modelov medzi najpoužívanejšie. Jadrom celého modelu je frázová tabuľka, ktorá je vytvorená napríklad nasledujúcim postupom. Najprv je nutné definovať konzistenciu frázy zo zarovnaním slov:

$$\begin{aligned}(\bar{e}, \bar{f})\text{konzistentné s A} \quad &\Leftrightarrow \\ \forall e_i \quad &\in \quad \bar{e} : (e_i, e_f) \in A \;\Rightarrow f_i \in \bar{f} \\ \wedge \forall f_i \quad &\in \quad \bar{f} : (e_i, e_f) \in A \;\Rightarrow e_i \in \bar{e} \\ \wedge \exists e_i \quad &\in \quad \bar{e}, f_i \in \bar{f} : (e_i, e_f) \in A \end{aligned} \tag{2.15}$$

Príklady konzistentných zarovnaní možno nájsť na obrázku 2.9. Vyobrazené nekonzistentné zarovnanie porušuje podmienku, že žiadny bod zarovnania nemôže ležať mimo daného páru fráz.

Obrázok 2.9: Konzistencia zarovnania slov [6]

Na základe definície konzistencie môžeme zostrojiť algoritmus, ktorý extrahuje všetky páry fráz z existujúceho zarovnania slov. Princípom je iterácia nad všetkými možnými frázami cieľového jazyka a hľadanie najkratšej konzistentnej frázy cudzieho jazyka. Avšak je nutné uvažovať nasledujúce skutočnosti:

- Pokiaľ neobsahuje fráza v cieľovom jazyku zarovnané slová tak ju neuvažujeme.
- Ak obsahuje nájdená najkratšia fráza cudzieho jazyka slová so zarovnaním mimo uvažovanej frázy, tak ju ignorujeme.

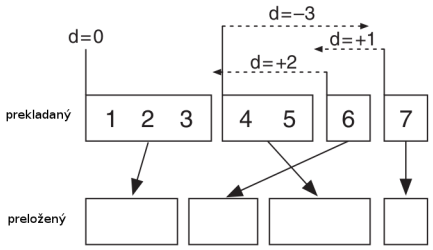
- Pokiaľ nájdená fráza susedí s nezarovnanými slovami, tak okrem nej pridáme aj frázy vzniknuté rozšírením minimálnej frázy o tieto slová.

Pravdepodobnosť výskytu danej dvojice fráz je možné určiť pomocou ich relatívnej frekvencie výskytu:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e}|\bar{f})}{\sum_{f_i} \text{count}(\bar{e}|\bar{f}_i)} \tag{2.16}$$

Princíp prekladu pozostáva z rozdelenia vstupného textu **f** na I fráz $\bar{f}_i$ a následný preklad každej z nich. Explicitne nie je určené toto rozdelenie, preto všetky možnosti rozdelenia sú rovnako pravdepodobné.

Zmena poradia slov je modelovaná pomocou jednoduchej penalizácie v závislosti od dĺžky presunu frázy. Situácia je znázornená na obrázku 2.10. Pre tento model sa nie je nutné učiť pravdepodobnosť presunu, ale závisí od vzdialenosti a parametru α podľa vzťahu $d(x) = \alpha^{|x|}$



Fráza	Prekladá	Pohyb	Vzdialenosť
1	1-3	začína na začiatku	0
2	6	preskakuje 4-5	+2
3	4-5	vracia sa pred 4-6	-3
4	7	preskakuje 6	+1

Obrázok 2.10: Zmena poradia slov v závisloti od vzdialenosti [6]

Typický model založený na princípe tabuľky fráz tak pozostáva z troch častí:

- Frázovej tabuľky $\phi(\bar{f}_i|\bar{e}_i)$.
- Modelu zmeny poradia slov d
- Modelu jazyka $p_{LM}(e)$

Nájdenie výsledného prekladu tak priamo závisí od súčinu čiastočných výstupov modelov, ktoré predstavujú jednotlivé časti:

$$e_{textnajlepia} = \arg\max_{e} \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi} d(\text{začiatok}_i - \text{koniec}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|e|} p_{LM}(e_i|e_1, \ldots, e_{i-1})^{\lambda_L M} \tag{2.17}$$

Pretože jednotlivé zložky môžu produkovať rôzne kvalitné výstupy, zavádzame váhy $\lambda_p hi, \lambda_d, \lambda_{LM}$, ktorými dokážeme zdôrazniť tie časti, s ktorými výstupom sme spokojný.

Zavedenie váh a následná formalizácia vyjadrenia modelu do podoby logaritmicko-lineárneho modelu umožňuje jednoducho pridávať ďalšie komponenty do výsledného prekladového systému.

Pravdepodobnosť prekladu v obidvoch smeroch Napriek tomu, že zvyčajne nám pre preklad stačí získať pravdepodobnosť v smere prekladu, môžu nastať okolnosti, keď dôjde k chybnému mapovaniu zriedkavej frázy cudzieho jazyka ku veľmi frekventovanej frázy cieľového jazyka. Tým pádom by pri následnom pokuse o preklad tejto zriedkavej frázy veľmi pravdepodobne došlo ku chybnému prekladu. Keďže ale máme možnosť rozšíriť náš systém o pravdepodobnosti prekladu obomi smermi, je možné tejto chybe zabrániť. Naviac model využívajúci oba smery prekladu, pokiaľ má vhodne zvolené váhy jednotlivých smerov, zvyčajne dosahuje lepšie výsledky ako model, ktorý využíva iba jeden smer.

Lexikálne váhy Opäť sa jedná o predchádzanie chybného prekladu ojedinelých fráz, ktoré boli ale zle naučené. V tomto prípade sa jedná o spôsob vyhladzovania. Pokiaľ máme k dispozícii pôvodné zarovnanie slov, z ktorého sme extrahovali frázy, tak môžeme vypočítať pravdepodobnosť lexikálneho prekladu ako:

$$\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{I} \frac{1}{|\{j : (i,j) \in a\}|} \sum_{\forall (i,j) \in a} t(e_i|f_i) \tag{2.18}$$

Podobne ako v prípade pravdepodobnosti prekladu fráz je vhodné doplniť oba možné smery mapovania a vhodne zvoliť váhy.

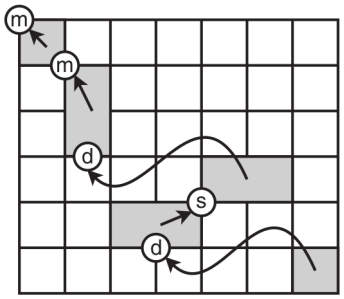
Penalizácia výslednej dĺžky prekladu Zatiaľ sme v systéme nemali priamy vplyv na výslednú dĺžku prekladu. Avšak model jazyka preferuje kratšie vety, pretože v takom prípade vstupuje do súčinu pravdepodobností menší počet n-gramov. Získať vplyv nad výslednou dĺžkou výstupu možno zavedením penalizácie slov ω, ktorá penalizuje každé vyprodukované slovo. Nastavením jej hodnoty môžeme preferovať dlhšie alebo kratšie vety.

Modelovanie zmeny poradia slov

Pokiaľ by nám nevyhovoval zatiaľ použitý veľmi jednoduchý model zmeny poradia slov, môžeme použiť model založený na pôvodnom zarovnaní slov. Existujú tri druhy zmeny poradia slov, ktoré tento model uvažuje a sú definované pomocou orientácie fráz (obrázok 2.11). Pre získanie tejto orientácie potrebujeme zachovať zarovnanie slov, z ktorého sme vytvorili dané frázy. Orientácia frázy je určená pomocou susediacich bodov a to tak, že:

- Monotónna je vtedy, ak existuje susediace zarovnanie vedľa ľavého horného bodu zarovnania.

- Zámena nastáva vtedy, ak existuje susediace zarovnanie vedľa pravého horného bodu zarovnania.
- Prerušené je vtedy, ak nie je ani monotónne ani zamenené.



Obrázok 2.11: Orientácia fráz: m - monotónna, s - zámena, d - prerušená [6]

Pre každý typ zarovnania spočítame jeho pravdepodobnosť podľa počtu výskytov danej frázy v danom zarovnaní:

$$p_o(\text{orientácia}|\bar{f},\bar{e}) = \frac{\text{počet}(\text{orientácia},\bar{e},\bar{f})}{\sum_o \text{počet}(o,\bar{e},\bar{f})} \tag{2.19}$$

Dekódovanie

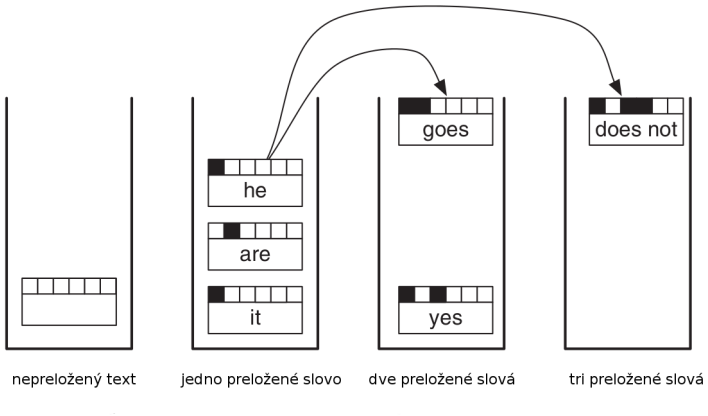
Úlohou dekódovania je nájsť preklad, ktorý dosahuje najlepšie skóre podľa hodnotenia použitého modelu. Tento problém je vlastne prehľadávaním stavového priestoru všetkých možných prekladov, z čoho jane vyplývam že táto úloha je časovo a pamäťovo náročná. Stavový priestor rastie exponenciálne v závislosti od dĺžky vety, ktorú máme preložiť. Z tohto dôvodu nie je možné prehľadávať všetky dostupné možnosti, ale na základe heuristických metód sa snažíme nájsť čo najlepšie riešenie. V skutočnosti ale nemáme zaručené, že sa k nemu dopracujeme, ale vhodným spôsobom prehľadávania sa k nemu môžeme priblížiť.

Samotný proces prekladu začína vytvorením prázdnej hypotézy. Pretože ešte neobsahuje žiadne preložené slová, jej čiastočné skóre je rovno 1. Následne vykonáme jeden krok prekladu - preložíme jednu z ešte nepreložených fráz a túto činnosť pokračujeme až kým nevyčerpáme všetky nepreložené frázy. Tým tvoríme vyhľadávací graf. Nakoniec zo všetkých hypotéz, ktorými sme preložili celú vetu vyberieme tú, ktorá dosahuje najvyššie skóre. Táto hypotéza je potom prehlásená za výsledný preklad.

Pretože vyhľadávací strom narastá do nezvládnuteľných rozmerov, nastupujú metódy ako eliminovať zbytočné kroky. Napríklad keď dve hypotézy preložili zhodné frázy pôvodného textu, nemá zmysel ďalej udržovať obe a tú s horším skóre vymažeme zo stromu. To, že majú rozdielne skóre je spôsobené tým, že mohli použiť rozdielne výstupné frázy alebo zmena poradia slov spôsobila nižšie ohodnotenie jazykového modelu.

Ďalšou možnosťou je organizácia hypotéz do košov podľa počtu preložených slov 2.12. Až dosiahne v nejakej kope počet hypotéz vopred daný počet, zmažeme najhoršie hypotézy

a pokračujeme ďalej. Inou možnosťou je zahadzovanie všetkých hypotéz, ktoré sú α krát horšie ako najlepšia hypotéza v danom koši.



Obrázok 2.12: Koše hypotéz [6]

I keď nám zahadzovanie nevhodných hypotéz uľahčuje hľadanie, je veľmi dôležité aby sme tak činili podľa správneho rozhodnutia. Vždy tam existuje riziko, že zo zahodenej hypotézy by eventuálne mohol po vyčerpaní slov na preklad vzniknúť najlepší výsledok. Preto je vhodné hodnotiť nielen súčasné čiastočné skóre prekladu, ale aj potenciálne získateľné skóre v budúcnosti. Jedným zo spôsobov tohto odhadu je využitie prekladu zatiaľ nepreložených slov, ale s tým, že neuvažujeme zmenu poradia slov ani hodnotenie modelom jazyka. Až takto dočasne preložíme všetky slová, najlacnejšie riešenie vrátime ako odhad budúcej ceny.

Samozrejme môžme použiť aj iné spôsoby efektívneho prehľadávania stavového priestoru, napríklad A* vyhľadávanie. Avšak musíme vhodne zvoliť potrebné hodnotiace funkcie. V prípade A* ale nemáme zaručené ukončenie v polynomiálnom čase a tiež hodnotenie budúceho zisku nesmie podhodnotiť odhad. Čím sme nútení byť viac opatrní a táto funkcia bude produkovať menej realistické hodnoty.

2.2.4 Modely založené na stromoch

Z jazykového hľadiska možno vety reprezentovať syntaktickými stromami a ich využitie pre k prekladu bolo skúmané už dlhšie. Ale až do nedávna systémy, ktoré ich používali, dosahovali horšie výsledky ako systémy využívajúce tabuľky fráz.

Gramatiky, ktoré popisujú tieto pravidlá vychádzajú z bezkontextových gramatík, kde môže byť len jeden non-terminál na pravej strane pravidla. Ľavá strana pravidiel môže obsahovať ľubovoľný počet terminálov a non-terminálov.

Bezkontextová gramatika je rozšírená na tzv. synchrónnu gramatiku, ktorá umožňuje zachytiť rozdiely medzi jazykmi.

Napr. frázy s podstatnými menami majú inú štruktúru v angličtine ako francúzštine. Pravidlo má teda tvar

$$NP \rightarrow DET_1 NN_2 JJ_3 | DET_1 JJ_3 NN_2$$

kde prvá časť pravej strany predstavuje francúzsky a druhá časť anglický tvar. Samotný preklad slov možno vyjadriť pravidlom pre priamy preklad ako

$$N \rightarrow maison | house$$

Ku každému pravidlu je priradené skóre vyjadrujúce pravdepodobnosť jeho použitia. Preklad dvojice viet je teda možné ohodnotiť podľa vzťahu

$$SCORE(TREE, E, F) = \prod_i RULE_i \tag{2.20}$$

Tieto pravidlá je možno získať aj bez doplnenia syntaktických informácií do viet, teda priamo z paralelného korpusu. V niektorých dvojiciach jazykov dosahujú lepšie výsledky ako tabuľky fráz.

Podobne ako pri tvorbe tabuľky fráz vychádzame zo zarovnania slov pre daný pár viet a extrahujeme všetky konzistentné frázy. Tieto frázy priamo vytvoria pre preklad tvaru

$$Y \rightarrow \bar{f} | \bar{e}$$

Z nich postupným zovšeobecňovaním vytvoríme komplexnejšie pravidlá, ktoré už budú obsahovať non-terminály i na pravej strane. Pokiaľ nahradíme non-terminálov slová, ktoré prerušujú frázu, tak implicitne zavedieme pravidlá pre zmenu poradia slov vo výslednom preklade.

Extrahujeme všetky páry fráz, ktoré sú konzistentné zo zarovnaním slov (podľa **??**) a rekurzívne dopĺňame pravidlá pokiaľ spĺňajú podmienku:

$$\begin{aligned} \text{pokiaľ} (\bar{e}, \bar{f}) \in P \quad &\wedge \quad (\bar{e}_{SUB}, \bar{f}_{SUB}) \in P \\ &\wedge \quad \bar{e} = \bar{e}_{PRE} + \bar{e}_{SUB} + \bar{e}_{POST} \\ &\wedge \quad \bar{f} = \bar{f}_{PRE} + \bar{f}_{SUB} + \bar{f}_{POST} \\ &\wedge \quad \bar{e} \neq \bar{e}_{SUB} \wedge \bar{f} \neq \bar{f}_{SUB} \end{aligned}$$

$$\text{pridaj } (e_{PRE} + X + E_{POST}, f_{PRE} + X + f_{POST}) \text{ do } P \tag{2.21}$$

Nevýhodou tohto prístupu je až exponenciálne množstvo hierarchických pravidiel z jednej dvojice viet, keďže môže dôjsť k mapovaniu ľubovoľnej podmnožiny slov vety. V praxi je tvorba nových pravidiel obmedzená pravidlami:

- najviac dva non-terminálne symboly
- 1 až 5 slov v páre
- vynechanie maximálne 15 slov
- non-teminály nie sú vedľa seba v oboch jazykoch

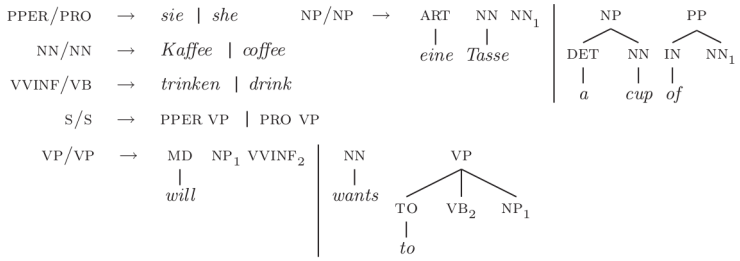
Dekódovanie

Priebeh prekladu vety je odlišný od modelov založených na tabuľke fráz. Nemôžeme budovať preklad postupne, pretože pravidlá obsahujú medzery. Pre preklad preto použijeme upravený algoritmus syntaktickej analýzy zdola hore, ktorý využije skutočnosť, že pravidlá obsahujú práve jeden non-terminálny symbol na ľavej strane.

Hlavnou myšlienkou je postupná aplikácia pravidiel začínajúc najkratšími. Po ich vyčerpaní sa aplikujú dlhšie pravidlá, až do pokrytia celej vety. Aplikovateľné pravidlá sa vykonajú a zaznamenajú do grafu. Pretože pravidlá už obsahujú preklad, tak každá položka tohto grafu obsahuje čiastočný preklad vety.

Hlavným problémom tohto riešenia je množstvo generovaných položiek, čo kladie nároky na organizáciu grafu a čo najskoršie zmazanie tých, ktoré už nevyhovujú.

Situáciu ilustrujú obrázky 2.13, ktorý obsahuje použité gramatické pravidlá a 2.14, ktorý zobrazuje postup ich aplikovania.



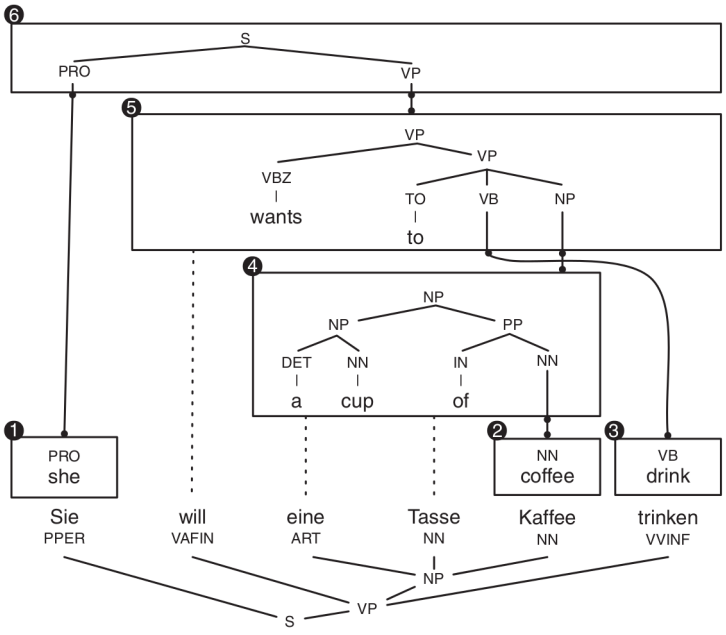
Obrázok 2.13: Gramatické pravidlá pre preklad [6]

Na vstupe máme nemeckú vetu „Sie will eine Tasse Kaffee trinken“. Najprv sme aplikovali pravidlá, ktoré pokrývajú jedno slovo (1-3). Následne už nemáme inú možnosť ako aplikovať dlhšie pravidlá. Pravidlo aplikované v 4 kroku pokrýva „eine Tasse Kaffee NN“, kde sa už vyskytuje prvý non-terminál z predošlých krokov. Až získame celý strom, môžeme vyprodukovať jeho preklad.

V praxi je celý problém zložitejší. Prehľadávanie stavového priestoru vyžaduje, podobne ako pri dekódovaní pomocou tabuliek fráz, rozumnú metodiku orezávania neefektívnych vetví. Taktiež je nutné zvýšiť efektivitu vyhľadávania

2.3 Modelovanie jazyka

Aby sme mohli určiť pravdepodobnosť s akou sa daná postupnosť slov môže vyskytnúť v jazyku, je potrebné poznať jeho model. Najrozšírenejšou metódou je využitie n-gramov, v uvedenom príklade tri-gram:



Obrázok 2.14: Výsledný strom po aplikovaní pravidiel [6]

$$\begin{aligned} p(\mathbf{e}) &= p(e_1, e_2, \ldots, e_n) \\ &= p(e_1)p(e_2|e_1)\cdots p(e_n|e_1, e_2, \ldots, e_{n-1}) \\ &\simeq p(e_1)p(e_2|e_1)\cdots p(e_n|e_{n-2}, e_{n-1}) \end{aligned} \tag{2.22}$$

Tento spôsob modelovania namiesto výpočtu pravdepodobnosti celej sekvencie slov určuje výslednú pravdepodobnosť pomocou pravdepodobností výskytu nasledujúceho slova na základe obmedzeného počtu predchádzajúcich slov. Jedná sa teda o Markovovu reťaz, kde počet uvažovaných predchádzajúci slov určuje stupeň modelu. Uvažovaná Markovova domnienka o nezávislosti slova dlhšej histórii neplatí, ale umožňuje nám vytvoriť model, ktorého výsledky sú dostačujúce.

Pri výpočte pravdepodobnosti sa v praxi často stretávame s prípadmi, keď sme danú postupnosť ešte nikdy nevideli. Podľa uvedeného vzťahu by tak pravdepodobnosť jej výskytu bola 0, čo nám nevyhovuje.

Jedným z riešením je pripočítať k všetkým výskytom pevne danú hodnotu α.

$$p = \frac{c + \alpha}{n + \alpha v} \tag{2.23}$$

Túto hodnotu je možné určiť v závislosti od trénovacích dát, aby sme dosiahli čo najlepšiu zhodu s výskytom v testovacích dátach.

Iným riešením je použite vyhladzovania výsledkov, kde sa na základe zaznamenaného počtu výskytov n-gramov v korpuse snažíme lepšie odhadnúť ich skutočný očakávaný počet

výskytov v jazyku. Good-Turingovo vyhladzovanie je definované vzťahom

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \tag{2.24}$$

v ktorom N_r vyjadruje, koľko n-gramov sa vyskytlo práve r krát v korpuse. Výsledný počet výskytov r^* je blízky skutočnému počtu. Pre vyššie hodnoty r často dosahuje $N_r = 0$ a preto je potrebné buď ponechať získanú hodnotu alebo aproximovať hodnotu pomocou jej okolia.

Rozumným využitím rôznych vlastností n-gramov, teda robustnosť nižších stupňov a zachytenie kontextu pomocou vyšších umožňuje interpolácia.

Príkladom je lineárna interpolácia

$$\begin{aligned} p(w_3|w_1,w_2) &= \lambda_1 p_1(w_3) \\ &\quad \times \lambda_2 p_2(w_3|w_2) \\ &\quad \times \lambda_3 p_3(w_3|w_1,w_2) \end{aligned} \tag{2.25}$$

kde jednotlivé váhy λ_n určujú vplyv jednotlivých n-gramov.

2.4 Metriky kvality prekladu

Táto časť je venovaná spôsobom hodnotenia kvality prekladu. Sumarizuje zadanie hodnotenia ich automatizovaným spôsobom hodnotenia a porovnávania kvality prekladu.

Ohodnotiť kvalitu prekladu je zložité. Na rozdiel od mnohých iných úloh spracovania jazyka, napríklad rozpoznávanie reči, zvyčajne neexistuje jednoznačne definovaný požadovaný výstup. Už pri aj pri veľmi krátkych a teda na oko jednoduchých vetách, viacerí prekladatelia formulujú svoje výsledky svojho prekladu rozdielne. Taktiež je veľmi zložité prehlásiť niektorý preklad za správny, pretože samotný pojem správnosť prekladu nie je presne definovaný a jeho výklad je do značnej miery subjektívny a závisí od jazykového citu hodnotiaceho.

Preto sa pri hodnotení prekladu využívajú nasledujúce dve vlastnosti:

- Plynulosť prekladu: vyjadruje nakoľko výsledný text odpovedá správnej skladbe vety jazyka.
- Adekvátnosť prekladu: určuje ako bol zachovaný zmysel vety.

Manuálne hodnotenie prekladu býva vo forme dotazníku, v ktorých sa hodnotia jednotlivé časti prekladu. Prípadne je hodnotiaci požiadaný o porovnanie dvoch systémov. Následne sa viacero výsledkov spriemeruje. Tieto posudky sú kvalitné, ale je nutné ich opakovať pri každom novom preklade, čím sa stávajú nedostupnými pri často opakovaných požiadavkách hodnotenia, napr. počas vývoja prekladového systému.

Riešením je automatizované hodnotenie prekladu, ktoré bude spĺňať nasledujúce požiadavky:

- Je lacné: nie je problémom ho spúšťať často a výsledky sú dostupné v krátkom čase.
- Je konzistentné: viacero hodnotení používajúcich rovnakú metriku dosiahne rovnaké výsledky.
- Je správne: výsledky by mali byť podobné hodnoteniu človekom.

Automatizovateľné spôsoby hodnotenia prekladu sú založené väčšinou na princípe porovnávania viet testovaného výstupu a jeho referenčných prekladov.

2.4.1 Chybovosť slov

Chybovosť slov (Word Error Rate) [6] je jednou z prvých používaných automatizovaných metrík. Vychádza z Levensteinovej vzdialenosti 2.26, ktorá hodnotí podobnosť slov na základe počtu a váh operácií, vykonaním ktorých dosiahneme zhodu týchto dvoch slov. Preto býva označovaná aj ako editačná vzdialenosť. Povolenými operáciami sú zhoda, nahradenie, vloženie a zmazanie. Pre výpočet chybovosti slov určujeme namiesto písmen slova počet operácií slov vo vetách.

$$WER = \frac{\text{zámen} + \text{vložení} + \text{zmazaní}}{\text{dĺžka referencie}} \tag{2.26}$$

Hlavnou nevýhodou tejto metriky je vyžadovanie poradia slov vo vete, ktoré nie je vždy až také dôležité, hlavne v jazykoch s voľnejšou skladbou viet (napríklad slovenčina).

2.4.2 BLEU

Aktuálne asi najznámejšou a stále veľmi populárnou metrikou je BLEU (Bilingual Evaluation Understudy)[11]. Vychádza z podobného princípu ako predchádzajúca metrika, avšak uvažuje zhodu n-gramov medzi prekladom a jeho referenciami. Samotné porovnávanie zhody n-gramov je nezávislé na ich pozícii vo vete a teda prehodenie slovosledu nemá až taký vplyv na výslednú hodnotu. Naviac umožňuje definovať viacero referenčných prekladov, čím eliminuje individuálny štýl prekladateľa.

Najjednoduchším príkladom n-gramu je samostatné slovo. Pokiaľ uvažujeme len celkový počet zhodných slov voči celkovej dĺžke prekladu, tak by sa mohlo stať, že malý počet vhodne zvolených slov by dosiahol vysoké skóre napriek jeho nezmyselnosti. Príkladom môže byť výsledný preklad „áno sme áno sme áno“, ktorý by dosiahol vysoké skóre, pretože všetky jeho slová sa nachádzajú v referencii. Ale my by sme skôr očakávali nízke skóre, pretože referenčným prekladom je „áno, prišli sme v poriadku“.

Riešením je zaviesť maximálny počet výskytu slov, ktorý je určený všetkými referenčnými prekladmi. V predchádzajúcom príklade by bol teda použitý len jeden výskyt slov „áno“ a „sme“.

Ďalším problémom, ktorý BLEU musí uvažovať, je celková dĺžka prekladu. Orazanie n-gramu zamedzuje nadmerné používanie rovnakých slov a slovných spojení, ale nevynucuje zachovanie očakávanej dĺžky prekladu. Riešením je penalizácia veľkej stručnosti prekladu. Keďže BLEU uvažuje viacero referenčných prekladov, ktoré môžu mať rôzne dĺžky, za efektívnu dĺžku je považovaná tá referenčná dĺžka, ktorá má je najviac podobná dĺžke prekladu. Napríklad pre referencie dĺžky 5,9,11 a preklade s dĺžkou 8 je za referenčnú dĺžku považovaná hodnota 9. Aby sa zabránilo prehnanej penalizácii odchýlok krátkych viet, penalizácia stručnosti je počítaná pre celý korpus.

Počet zhôd slov určuje adekvátnosť prekladu, narastajúca dĺžka n-gramov zvyšuje požiadavku na plynulosť prekladu. Vzťah pre výpočet skóre pre n-gram 2.27.

Autori metriky BLEU ukázali, že všetky n-grami sú schopné rozlíšiť medzi dobrým a zlým prekladom. Avšak je nutné umožniť porovnanie kvality medzi dvomi prekladmi. Keďže hodnota presnosti klesá približne exponenciálne s narastajúcou veľkosťou N, je vhodné použiť logaritmickú schému pre zhrnutie presnosti rôznych n-gramov 2.28.

$$\text{skóre}_n = \frac{\sum_{K \in \{\text{Kandidáti}\}} \sum_{n-gram \in K} \text{Výskyt-orezaný (n-gram)}}{\sum_{K' \in \{\text{Kandidáti}\}} \sum_{n-gram' \in K'} \text{Výskyt (n-gram')}} \tag{2.27}$$

$$\log BLEU = \min(1 - \frac{\text{dĺžka výstupu}}{\text{referenčná dĺžka}}, 0) + \sum_{n=1}^{N} w_n \log p_n \tag{2.28}$$

Výhrady voči BLEU

V článku [2] jeho autori poukázali na fakt, že veľmi zdôrazňovaná vlastnosť korelácie medzi hodnotu BLEU a subjektívnym hodnotením človekom neplatí vždy. Tento rozdiel nastáva najmä pokiaľ sú prekladové systémy založené na rozličných prístupoch, čo je častým príkladom použitia automatického hodnotenia, najmä pokiaľ chceme určiť, ktorý z prístupov je pre danú dvojicu jzaykov vhodnejší.

Hlavným problémom sú permutácie a substitúcie, ktoré BLEU umožňuje pri zachovaní podobného skóre. Človek ich ale zvyčajne ohodnotí rozdielne. V prípade b bi-gramov a dĺžke prekladu vety k je týchto možností až $(k - b)!$

BLEU tiež prisudzuje všetkým slovám rovnakú váhu a považuje substitúciu slov tvoriacich kontext za rovnakú ako substitúciu funkčných slov. Rozdiel medzi kontextom prekladu a referencie sú ale dôležitým hodnotiacim prvkom pri hodnotení prekladu človekom.

V neposlednom rade samotná hodnota skóre nemá sama o sebe vypovedaciu hodnotu,

pretože veľmi závisí od domény text, dĺžky korpusu, počtu dostupných referencií a dokonca aj od použitého spôsobu tokenizácie.

2.4.3 Meteor

Je taktiež metrikou, ktorá hodnotí kvalitu prekladu podľa zhody medzi prekladom a jeho referenciami [3]. Snaží sa vyvarovať väčšine problémov, ktorými bol kritizovaný BLEU a preto táto metrika umožňuje využiť vedomosti o jazyku a použiť ich pri vyhodnocovaní zhody medzi prekladom a referenciami. Tak je možné zachytiť preklad pomocou synoným a parafráz, ktoré BLEU vždy označil za chybu. Veľkou nevýhodou však je, že tieto jazykové poznatky musia existovať, čo ešte stále neplatí pre väčšinu jazykov. Podobne ako BLEU umožňuje porovnať preklad s viacerými referenciami, čo značne zmierňuje preferovanie jedného a toho istého prekladateľského štýlu. V prípade, že existuje viac referencií, dôjde k vyhodnoteniu všetkých z nich a za výslednú hodnotu je považované najlepšie získané skóre.

Vyhodnocovanie zhody medzi prekladom a referenciou pomocou porovnania na:

- Presnú zhodu (Exact): tvar slova sa zhoduje.
- Koreň (Stem): korene slov sa zhodujú.
- Synonymum: slová sú si navzájom synonymami.
- Parafráza: slová sú si navzájom parafrázami.

Okrem porovnania na presnú zhodu všetky ostatné funkcie vyžadujú vedomosti o jazyku.

Po identifikovaní párov sú tieto ohodnotené podľa nasledujúcich kritérií, ktoré sú zoradené zostupne:

1. Každé slovo v každej vete je pokryté najviac jedným slovom.
2. Maximalizácia počtu pokrytých slov v oboch vetách.
3. Minimalizovanie počtu zhlukov - súvislých pokrytí zachovávajúcich rovnaké poradie v oboch vetách.
4. Minimalizovanie sumy vzdialeností medzi nájdenými zhodami v oboch vetách.

Tým je určené konečné zoradenia a je možné počítať jeho skóre. Najprv identifikujeme funkčné slová a slová tvoriace obsah v preklade (h_f, h_c) a referencii (r_f, r_c). Pre každý použitý spôsob porovnania (m_i) spočítame výskyt zhôd pre funkčné slová a obsah tvoriace slová zvlášť. Získame tak hodnoty pre preklad $(m_i(h_c), m_i(h_f))$ a referenciu $(m_i(r_c), m_i(r_f))$, ktoré použijeme pre výpočet presnosti P 2.30 a použitia (recall) R **??**. Váhou δ určujeme pomer dôležitosti medzi funkčnými a obsah tvoriacimi slovami. Následne vypočítame harmonický priemer 2.31.

$$P = \frac{\sum_i w_i(\delta m_i(h_c) + (1-\delta)mi_i(h_f))}{\delta|h_c| + (1-\delta)|h_f|} \tag{2.29}$$

$$R = \frac{\sum_i w_i(\delta m_i(r_c) + (1-\delta)mi_i(r_f))}{\delta|r_c| + (1-\delta)|r_f|} \tag{2.30}$$

$$F_{mean} = \frac{P \times R}{\alpha P + (1-\alpha)R} \tag{2.31}$$

Aby výsledné skóre záviselo aj od dodržania slovosledu a použitia správneho počtu slov, zavedieme penalizáciu fragmentácie 2.32. Váhu tejto penalizácie vo výslednom skóre určuje parameter γ, samotný pomer počtu zhlukov a celkového počtu zhôd ovplyvňuje parameter β.

$$Pen = \gamma \left(\frac{ch}{m}\right)^{\beta} \tag{2.32}$$

Nakoniec použijeme hodnoty harmonického priemeru a penalizácie pre výpočet výsledného skóre S:

$$S = (1 - Pen)F_{mean} \tag{2.33}$$

Parametre α, β, γ, δ a váhy jednotlivých spôsobov hľadania zarovnania slov $w_i \dots w_n$ je nutné nastaviť v závislosti na cieľovom jazyku. Je potrebné ich určiť tak, aby bola maximalizovaná zhoda s manuálnym hodnotením.

Samotný systém Meteor je voľne dostupný k stiahnutiu[1] a je vydaný pod licenciou GNU LGPL.

[1] http://www.cs.cmu.edu/~alavie/METEOR/index.html

Kapitola 3

Návrh systému

Táto kapitola obsahuje návrh systému pre preklad z češtiny do slovenčiny. Kapitola sa ďalej člení na kratší popis zdrojov dvojjazyčných textov a prehľad použitých nástrojov pre zostavenie systému.

3.1 Zdroje dvojjazyčných textov

Výslednú kvalitu prekladu jazykových modelov priamo obmedzujú zdroje ich dát. Nie je možné preložiť slovo s ktorým sme sa ešte nestretli a tiež nie je možné správne formulovať výstupný text, pokiaľ nepoznáme dostatočný počet príkladov použitia. Medzi základné zdroje dát patria dokumenty, ktoré existujú v oboch jazykoch a obsahom sa veľmi nelíšia. Pretože každý text je zasadený do istého kontextu, pre obecný systém je nutné skombinovať viacero dostatočne rozsiahlych zdrojov rôznych oblastí.

Keďže oba jazyky sú oficiálnymi jazykmi Európskej únie, je pre ne výborným zdrojom textov legislatíva EU. Existujú dva korpusy, ktoré túto legislatívu spracúvajú rozdielnym spôsobom.

JRC-Acquis Communautaire JRC-Acquis [13] je najväčším zdrojom viacjazyčných legislatívnych dokumentov pochádzajúcich z Európskej komisie. Samotné zarovnanie textov prebehlo automaticky pomocou nástrojov Vanilla a Hunalign.

DGT-TM DGT Multilingual Translation Memory [12] je výsledkom spracovania legislatívnych textov EU do podoby ustálených fráz (Translation Unit). Z týchto fráz nie je možné zrekonštruovať pôvodný text, tieto frázy majú za cieľ eliminovať duplicitu prekladu, či už z hľadiska zbytočnej práce, alebo zmene formy. Jedná sa o útržky viet, prípadne celé vety. Výsledné zarovnanie fráz bolo zväčša manuálne skontrolované. Pred samotným zarovnávaním boli texty vyčistené od častí s minimálnou prekladateľskou hodnotou.

OPUS OPUS[15] je kolekciou textov získaných z Internetu, ktoré sú automaticky spracované a zarovnané. Neprebieha tu žiadna manuálna korekcia výsledkov. Celý tento systém je postavený na Open Source produktoch a je dostupný voľne k stiahnutiu. Pre dvojicu čeština-slovenčina sú dostupné zarovnané texty vytvorené zo zdrojov Open Subtitles[1], EMEA (European Medicine Agency), ECB (Európskej centrálnej banky), projektu KDE, textu Európskej ústavy a PHP.

Open Source projekty Tieto projekty majú bohatú užívateľskú základňu a snažia sa prispôsobiť potrebám svojich užívateľov. Existuje ale veľký nepomer medzi preloženými programami do slovenčiny a češtiny, ktorý je v neprospech slovenskej lokalizácie. Je to spôsobené veľkosťou cieľovej skupiny, dostupnosti prekladateľov a faktu, že existujúca lokalizácia pre češtinu je dostačujúca pre používateľov, ktorí rozumejú obom jazykom.

Jedným z veľkých zdrojov textov je projekt KDE[2], ktorý obsahuje jednak reťazce vyskytujúce sa v programoch a tiež dokumentáciu k nim. Z hľadiska vhodnosti pre účely automatického učenia prekladového modelu viac vyhovuje dokumentácia, táto je ale v prípade čestiny a slovenčiny skoro úplne ignorovaná.

V prípade systémov pre správu obsahu akými sú Joomla[3] alebo Drupal [4] kde sú preložené všetky dostupné texty avšak v týchto prípadoch ich nie je až tak veľa ako pre KDE. Komunita okolo jazyka PHP[5] je taktiež zdrojom textov, avšak v tomto prípade ich je žalostne málo.

Knihy Pre prípad jazykov čestina-slovenčina neexistuje dostatočné množstvo voľne šíriteľnej literatúry. Projekt Gutenberg[6] obsahuje len jednu jedinú knihu v slovenskom jazyku.

Slovníky Podobne v prípade kníh, ani slovníky dvojice čeština - slovenčina nie sú dostupné ako voľne šíriteľné. Existuje ale viacero komerčných riešení.

3.2 Použité nástroje

Pretože sú voľne dostupné už hotové systémy na rozhodli sme sa ich využiť. Asi najznámejším a najpoužívanejším systémom je implementujúci strojový preklad je voľne dostupný Moses [7]. Tento systém je založený na trénovaní modelov z paralelných dát, ktoré musia byť vo formáte navzájom zarovnaných viet. Dané vety slúžia k naučeniu fráz. Okrem modelov založených na tabuľke fráz sa dokáže Moses naučiť aj hierarchické modely fráz a taktiež syntaktické modely.

[1] http://www.opensubtitles.org/
[2] http://i18n.kde.org/
[3] http://www.joomla.org
[4] http://www.drupal.org
[5] http://www.php.net/
[6] http://www.gutenberg.org/

Samotný Moses sa skladá z kolekcie nástrojov pre učenie systému a dekodéru. Táto kolekcia nástrojov slúži pre zefektívnenie jednotlivých krokov učenia, ktoré využívajú externé nástroje. Jadrom je skript train-model.perl, ktorý postupne vyvoláva vykonanie všetkých deviatich krokov:

1. Príprava dát.
2. Beh (M)GIZA.
3. Zarovnanie slov.
4. Tvorba lexikálnej prekladovej tabuľky.
5. Extrakcia fráz.
6. Hodnotenie fráz.
7. Tvorba modelu pre zmenu poradia slov.
8. Tvorba generovacieho modelu.
9. Vytvorenie konfiguračného súboru.

Samotný systém Moses nevytvára tabuľku zarovnaní slov, ale namiesto toho využíva služby programu Giza++, prípadne voliteľne program MGIZA.

GIZA++ a MGIZA Oba programy implementujú IBM Modely, ktoré, ako bolo spomenuté v druhej kapitole, sú často používané pre tvorbu zarovnania slov. Pôvodný program GIZA++[10] bol rozšírený o podporu viacvláknového behu na viacjadrových procesoroch. Pokiaľ máme k dispozícii taký počítač, je vhodné uvažovať o preferovaní programu MGIZA[5], čím skrátime dobu časovo veľmi náročnej fázy hľadania zarovnaní medzi slovami. Pretože sme presne tento prípad, tak pre nájdenie zarovnania slov použijeme program MGIZA.

Modelovanie jazyka Podobne ako tvorba zarovnania slov, ani modelovanie jazyka nepatrí medzi kompetencie systému Moses. Samotný systém je dodávaný s KenLM[7], ale je možné použiť aj ďalšie tri modely: RandLM, IRST a SRI. Pre účely tejto práce sme sa rozhodli využiť model SRI[14].

Použitie modelov V systéme Moses existujú dva dekodéry: moses a moses_chart. Program moses implementuje dekodér modelov založených na tabuľkách fráz, moses_chart slúži pre dekódovanie hierarchických modelov. Nami navrhovaný systém bude potenciálne využívať oba prístupy. Dôvodom je snaha zistiť, ktorý prístup je vhodnejší pre túto dvojicu

[7] http://kheafield.com/code/kenlm/

jazykov. Zároveň sa berie dôraz na časové a pamäťové nároky. V tomto navrhnutom systéme neplánujeme používať značkovač syntaxe, všetky informácie pre naučenie budeme čerpať z textu korpusu.

Ako už bolo spomenuté, samotný beh učenia modelu riadi skript train-model.perl. Po jeho ukončení je zapísaný konfiguračný súbor modelu, ktorý obsahuje cesty k jednotlivým modelom a preddefinované váhy jednotlivých modelov. Tieto váhy nie sú optimálne, preto je nutné spustiť optimalizáciu váh. Skript mert-moses.pl slúži k optimalizácii týchto váh. Okrem neho potrebujeme sadu textov a ich správne preklady. Táto sada by nemala byť veľmi veľká, lebo skript opakovane spúšťa preklad pomocou aktuálnej konfigurácie a po skončení prekladu vyhodnotí zmeny dosiahnutého skóre. Na základe nich nastaví nové váhy jednotlivých modelov. Pokiaľ by bola testovacia sada príliš veľká, beh optimalizácie váh by bol časovo veľmi náročný.

Po skončení optimalizácie je systém pripravený na porovnanie s existujúcimi systémami. Pretože dekodér umožňuje špecifikovať viacero modelov pre preklad, bude zaujímavé sledovať ako pridanie ďalšieho modelu ovplyvní výsledné skóre.

Kapitola 4

Realizácia systému

Táto kapitola poukazuje na realizovanú činnosť pri tvorbe prekladového systému. Tiež zaznamenáva prevedené experimenty zo systémom a jeho porovnanie s dvomi reálnymi prekladovými systémami.

4.1 Príprava dvojjazyčných textov

Zo zdrojov spomínaných v predchádzajúcej kapitole sme nakoniec využili pripravené korpusy Acquis Communautaire, EUConst a PHP. Korpusy boli distribuované spolu so vzájomným zarovnaním viet, preto bolo potrebné extrahovať vzájomne zarovnané dvojice viet.

Korpus vytvorený z titulkov k filmov sme sa rozhodli nepoužiť, pretože zbežná kontrola kvality prekladu nedopadla dobre. Väčšina titulkov mala značne posunutý význam a príliš voľný preklad, z ktorého by nebolo možné vytvoriť vhodný model.

Okrem spomenutých hotových korpusov sme použili aj nami vytvorené podľa nasledujúceho postupu.

Príprava korpusu zo slovníkov Bol nám umožnený prístup k slovníkom umiestneným na sieťovom disku minerva1[1]. V danom adresári sa nachádza niekoľko prekladových slovníkov, avšak len pre francúzsky, ruský a španielsky existovali vo variantách pre slovenčinu i češtinu. Slovníkové dáta boli extrahované do formátu tabfile, v ktorom je na každom riadku uvedený jeden výraz a jeho možné preklady. Na základe spoločného prekladu do jedného z týchto troch jazykov boli spojené preklady české a slovenské významy. Záznam po spojení vyzeral ako:

```
přijít bouřka \t zatiahnuť sa\nzamračiť sa
```

[1] /mnt/minerva1/nlp/projects/dicts2lmf

Vo väčšine prípadov sa líšili počty slovenských a českých výrazov. Preto boli chýbajúce výrazy pred ďalším spracovaním nahradené prázdnym reťazcom. Pre každý riadok výsledného spojenia výrazov sa určila hodnota Levensteinovej vzdialenosti dvojice medzi českými a slovenskými prekladmi. Následne sa postupne odoberali dvojice, ktoré si boli najbližšie, až kým sme nevyčerpali všetky dvojice, pre ktoré existovali aj český aj slovenský výraz. Pretože sme predtým zarovnali počty výrazov prázdnymi reťazcami, tak nám buď neostal žiadna dvojica, alebo jeden z výrazov bol prázdny reťazec. Dôvodom, prečo sme vyberali dvojice podľa vzdialenosti a nie lineárne alebo náhodne bol ten, že sme chceli aby viacslovný výraz bol, pokiaľ je to možné, preložený tiež viacslovným výrazom. Výsledný korpus slovníkov[2] vznikol zjednotením všetkých nájdených dvojíc.

Príprava korpusu z Joomla, Drupal a KDE súborov Pre prípravu korpusov z týchto troch zdrojov bolo potrebné extrahovať reťazce. Joomla[3] pre uloženie týchto reťazcov používa súbory formátu .ini, KDE[4] a Drupal[5] používajú .po súbory. Postup tvorby bol ale podobný. Najskôr bolo nutné získať reťazce. pretože sme začali spracovávaním .po súborov, tak sme aj .ini súbory transformovali do zjednodušenej podoby .po súborov. Väčšina pomocných lokalizačných súborov je indexovaná reťazcami tretieho, štandardného jazyka. Situácia bola teda podobná ako v prípade spracovania slovníkov.

Avšak niektoré reťazce nemuseli byť lokalizované do obidvoch jazykov. Preto sme explicitne kontrolovali či existujú oba preklady. Následne nastupovalo rozdelenie do viet. Lokalizované reťazce môžu byť zložené z viacerých viet, ale nástroje pre učenie sa prekladu z korpusu vyžadujú rozdelenie do viet. Preto sme nad každou dvojicou reťazcov (čeština-slovenčina) spravili kontrolu, či je ju možné rozdeliť do viet. Pokiaľ áno, tak oba reťazce boli rozdelené do viet a zarovnané. Pokiaľ došlo k tomu, že nebolo možné nájsť zarovnanie (počty viet boli rozdielne) tak sa celý pôvodný pár zahodil. Zarovnané páry boli nakoniec zapísané do odpovedajúcich súborov.

Dohromady bolo použitých celkovo sedem korpusov. Dva z nich predstavujú právnickú doménu, štyri predstavujú doménu komunikácie programu s užívateľom a jeden korpus je založený na prekladovom slovníku. Pred učením boli korpusy ešte vyčistené, tokenizované a zmenené do malého písma (lowercase). Z každého korpusu bol vytvorený model využívajúci tabuľku fráz a tiež aj model založený na stromových pravidlách. Jazykový model bol tiež vždy vytvorený pre odpovedajúci korpus.

4.2 Vyhodnotenie systému

Pre hodnotenie kvality prekladu bola použitá metrika BLEU, pretože pre slovenčinu nie sú dostupné doplňujúce informácie o jazyku, ktoré vyžaduje pre svoju správnu činnosť metrika Meteor.

[2] /mnt/minerva1/nlp/projects/mt_sk/korpusy/slovniky
[3] /mnt/minerva1/nlp/projects/mt_sk/korpusy/Joomla
[4] /mnt/minerva1/nlp/projects/mt_sk/korpusy/KDE
[5] /mnt/minerva1/nlp/projects/mt_sk/korpusy/Drupal

Existujú dve sady testovacích viet. Prvou je kompilácia pravidiel ochrany osobných údajov [6] a zmluvných podmienok[7] spoločnosti Google. Tieto texty boli vybrané pretože existuje aj český aj slovenský oficiálny preklad a sú vhodné pre otestovanie ako si s nimi poradia modely naučených na legislatívnych dokumentoch.

Druhou sadou sú lokalizačné texty CRM systému Wordpress[8], ktoré boli pripravené podobným spracovaním ako bolo použité pri tvorbe korpusu na základe lokalizačných textov Joomla alebo Drupal. Tieto texty tiež existujú v českej a slovenskej lokalizácii a sú vhodné najmä pre otestovanie modelov naučených na lokalizačných textoch.

Ako referenčné existujúce systémy boli použité Google translator[9] a Microsoft Bing[10]. Hodnoty BLEU skóre, ktoré dosiahli na testovacích textoch, sú uvedené v tabuľke 4.1. Zo systému Česílko[11], ktorý vyvinuli na na Karlovej Univerzite v Prahe, sa nám nepodarilo získať preklady vhodné k porovnaniu.

BLEU skóre	Google translator	Bing translator
Pravidlá a Zmluvné podmienky	2.45	0.82
Wordpress	5.05	6.35

Tabuľka 4.1: Dosiahnuté skóre BLUE na existujúcich on-line systémoch

Učenie nami navrhnutého systému prebiehalo pomocou skriptu dostupného so systémom Moses. Ako jazykový model bol použitý SRILM a väčšina paramterov bola ponechaná na prednastavených hodnotách. Po skončení učenia prebehla optimalizácia hodnôt váh, pričom za trénovacie dáta pre túto úlohu boli určené náhodným výberom z korpusu.

Prvým experimentom bolo otestovanie naučených systémov na jednotlivých testovacích sa dát. Tieto modely ešte neprešli optimalizáciou ale BLUE skóre dosiahli pomerne dobré. Výsledky modelov založených na tabuľke fráz sú uvedené v tabuľke 4.2.

BLEU skóre	Acquis	Drupal	Joomla	KDE	PHP	slovníky
Pravidlá a Zmluvné podmienky	1.24	0.64	0.49	0.86	0.24	0.87
Wordpress	7.25	6.89	5.9	9.44	2.58	8.87

Tabuľka 4.2: Dosiahnuté skóre BLUE neoptimalizovaných modelov založených na tabuľke fráz

Výsledky modelov založených na stromových pravidlách sú uvedené v tabuľke 4.3. Všetky systémy vykázali vyššiu hodnotu skóre BLUE v porovnaní s frázovými tabuľkami.

Ďalším krokom bola optimalizácia, po ktorej vyhodnotení sa ukázalo zhoršenie skóre BLEU4.4. Joomla dokonca dosiahla skóre 0 pre legislatívny dokument a PHP sa nepodarilo preložiť text vôbec.

[6] https://www.google.cz/intl/sk/policies/privacy/
[7] https://www.google.cz/intl/sk/policies/terms/regional.html
[8] http://wordpress.org/
[9] http://translate.google.com/
[10] http://www.microsofttranslator.com/
[11] http://quest.ms.mff.cuni.cz/cesilko/

BLEU skóre	Drupal	Joomla	KDE	slovníky
Pravidlá a Zmluvné podmienky	0.65	0.52	0.98	0.88
Wordpress	7.08	6.52	11.61	9.31

Tabuľka 4.3: Dosiahnuté skóre BLUE neoptimalizovaných modelov založených na stromových pravidlách

BLEU skóre	Acquis	Drupal	Joomla	KDE	PHP	slovníky
Pravidlá a Zmluvné podmienky	1.05	0.45	0	0.45	0	0.79
Wordpress	6.3	4.33	1.21	6.79	0	8.24

Tabuľka 4.4: Dosiahnuté skóre BLUE optimalizovaných modelov založených na tabuľke fráz

Možnou príčinou tohto problému je použitie náhodného výberu z trénovacích dát ako optimalizačné zadanie. Mohlo dôjsť k preučeniu a teda následnej degradácii kvality na „ostrých“ testovacích textoch.

Kapitola 5

Záver

V tejto práci čitateľ nájde zjednodušený prehľad často používaných metód automatického strojového prekladu. V druhej kapitole je oboznámený so typickými problémami a ich riešeniami ako sú zarovnanie viet, slov, pravdepodobnostné modelovanie prekladu a jazyka. Pretože táto práca využíva systém Moses ako hlavný stavebný prvok, boli predložené hlavne tie metódy, ktoré tento systém využíva.

Týmto sa dostávame k druhej časti zadania a tou bol návrh a implementácia systému pre preklad českých textov do slovenčiny. Návrh pozostával z uvedenia zdrojov dvojjazyčných textov, ktoré sú veľmi dôležitým stavebným prvkom štatistického prekladu. Bol popísaný postup vytvorenia nových korpusov založených nad lokalizačnými reťazcami Joomla, Drupal a KDE.

Text korpusov bol pred začiatkom učenia vyčistený od boli odstránené zbytočné symboly, prevedená tokenizácia a následný zmena textu do malých písmen. K značkovaniu vetných členov a druhov nedošlo, nebolo to pokladané za nutné. Z hľadiska použitých metód sme sa rozhodli naučiť dva spôsoby reprezentácie prekladových modelov, ktoré systém Moses umožňuje: tabuľka fráz a gramatické pravidlá. Pre schopnosť urýchliť spracovanie zarovnania slov na viac jadrových procesoroch sme sa rozhodli použiť MGIZA namiesto GIZA++. Použitým programom pre modelovanie jazyka bol SRI. Samotný proces učenia a následnej optimalizácie bol ponechaný na nástroje dodávané spolu so systémom Moses.

Treťou a poslednou časťou zadania bolo vyhodnotenie úspešnosti tohto modelu. Ako použitú metriku sme použili BLEU. Myšlienka metriky Meteor a jej prezentované výsledky sú sľubné, avšak jej závislosť na vedomostiach o jazyku ju robia ešte stále nedostupnou pre slovenčinu.

Výsledky porovnania navrhnutého systému boli síce v porovnaní s Google translator[1] a Bing translator[2] lepšie, avšak zbežná kontrola výstupu ukázala, že text je pomerne zle preložený. Ako testovacia sada textov bola, vzhľadom k prevažujúcim doménam korpusov,

[1] http://translate.google.com
[2] http://www.microsofttranslator.com/

vybraná sada právnických dokumentov a reťazce webového systému.

Prekvapujúco sa ukázalo, že optimalizácia zhoršila výsledky BLEU. Hypotézou je, že nebola vhodne zvolená sa trénovacích dokumentov a došlo k preučeniu, resp. prílišnej optimalizácii váh pre daný konkrétny text.

Vzhľadom k bohatosti oboch jazykov sa tiež ukázal nedostatok pokrytia slov. Z tohto hľadiska by bolo asi vhodnejšie učenie a preklad s využitím informácií o tvaroch slov.

Do budúcnosti by bolo asi vhodné vyskúšať alternatívne zarovnanie slov [8], ktoré na rozdiel od MGIZA nemusí prevádzať spätnú symetrizáciu na základe dvoch jednosmerných zarovnaní. Medzi vyzdvihované prednosti tohto prístupu patrí okrem menšej časovej náročnosti aj vyššia kvalita zarovnania.

Ďalším priestorom pre možné objavenie nových možností je použitie menej známych systémov strojový preklad cdec[3] a Joshua decoder [4]. Oba systémy sa zaoberajú modelovaním prekladu pomocou stromov a gramatík.

Pre zlepšenie pokrytého počtu slov by bolo vhodné rozšíriť korpusy o beletriu, čo častokrát nie je práve najjednoduchšie. Taktiež má beletria väčšie nároky na zarovnávanie viet, lebo prekladateľský štýl je občas voľnejší. Možno rozumnejším riešením by bola spolupráca s jazykovednými ústavmi, ktoré by už mali mať vytvorené národné korpusy oboch jazykov.

[3] http://cdec-decoder.org/index.php?title=Main_Page
[4] http://joshua-decoder.org/

Literatúra

[1] *History of machine translation* [[online]]. 22.05.2012. Dostupné na: <http://en.wikipedia.org/wiki/History_of_machine_translation>.

[2] Callison burch, C. a Osborne, M. Re-evaluating the role of BLEU in machine translation research. In *In EACL*. 2006. S. 249–256.

[3] Denkowski, M. a Lavie, A. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*. 2011.

[4] Gale, W. A. a Church, K. W. A program for aligning sentences in bilingual corpora. *Computational Linguistics*. 1993.

[5] Gao, Q. a Vogel, S. Parallel implementations of word alignment tool. In *In Proc. of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*. 2008.

[6] Koehn, P. *Statistical Machine Translation*. [b.m.]: Cambridge University Press, 2009. ISBN 0-521-87415-7.

[7] Koehn, P. *Statistical Machine Translation System. User Manual and Code Guide* [[online]]. 22.05.2012. Dostupné na: <http://www.statmt.org/moses/manual/manual.pdf>.

[8] Liang, P., Taskar, B. a Klein, D. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006. S. 104–111. HLT-NAACL '06.

[9] Manning, C. D. a Schütze, H. *Foundations of Statistical Natural Language Processing*. [b.m.]: MIT Press, 1999. ISBN 0-262-13360-1.

[10] Och, F. J. a Ney, H. Improved Statistical Alignment Models. In. Hongkong, China: [b.n.], October 2000. S. 440–447.

[11] Papineni, K., Roukos, S., Ward, T. et al. BLEU: a Method for Automatic Evaluation of Machine Translation. In. 2002. S. 311–318.

[12] Ralf, S., Eisele, A., Klocek, S. et al. DGT-TM: A freely Available Translation Memory in 22 Languages. In *Proceedings of the 8th international conference on Language Resources and Evaluation (LREC'2012)*. 2012.

[13] Ralf, S., Pouliquen, B., Widiger, A. et al. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*. 2006.

[14] Stolcke, A. SRILM—An extensible language modeling toolkit. In *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*. 2002. S. 901–904.

[15] Tiedemann, J. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In Nicolov, N., Bontcheva, K., Angelova, G. et al. (ed.). *Recent Advances in Natural Language Processing*. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009. S. 237–248. ISBN 978 90 272 4825 1.